IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Wolfgang Pfeifer	Art Unit	: 2194
Serial No.	: 10/811,063	Examiner	: Nathan E. Price
Filed	: March 25, 2004	Conf. No.	: 8061
Title	: COMMUNICATING MESSAGE WITH TYPE, OBJECT AND IDENTIFIERS FROM BUSINESS APPLICATION TO SERVICE APPLICATION		

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

Appellant files this brief on appeal under 37 CFR § 41.37, thereby perfecting the Notice of Appeal originally filed on July 2, 2008 after the Advisory Action mailed June 2, 2008 ("Advisory Action") and the final Office Action mailed February 5, 2008 ("Office Action").

(1) Real Party in Interest

SAP AG, the assignee of record in the present Application, is the real party in interest.

(2) Related Appeals and Interferences

There are no related appeals or interferences.

(3) Status of Claims

Claims 1-8 are pending in the application, with 1, 4, 6, and 7 being independent. All pending claims stand rejected, and all pending claims are on appeal.

(4) Status of Amendments

All amendments have been entered and no amendments are being submitted herewith.[1]

---

[1] The Advisory Action indicates that "[t]he proposed amendment(s) filed after a final rejection, but prior to the date of filing a brief, will not be entered because…(c) [t]hey are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal." Advisory Action, p. 2. However, Appellant respectfully submits that no amendments were made in the reply filed on May 5, 2008 in response to the final Office Action. Therefore, Appellant respectfully submits that, contrary to the Advisory Action's statement, all amendments have been entered and no amendments are being submitted herewith.

(5) Summary of Claimed Subject Matter

Independent claim 1 is directed to a computer system for identifying a target component in an apparatus that has components related in a hierarchy. *See Present Application*, *e.g.,* page 2, lines 3-4. The computer system may comprise a first computer (*such as* 900, page 5, lines 5-16) and a second computer (*such as* 901 or 902, page 5, lines 3-11), the second computer coupled to the first computer via a network. *See id.*, *e.g.*, page 5, lines 3-11. The first computer is operable to execute a first application (*such as* 201, page 8, lines 7-8) in which objects represent corresponding components by relating the objects in both a hierarchy identifying types of components and a different hierarchy identifying information associated with objects, wherein the type hierarchy and the object hierarchy identify the components in a first natural language. *See id.*, *e.g.*, page 8, lines 11-20. The first computer includes a message generator (*such as* 101, page 5, lines 13-15) operable to receive information relating to both the type hierarchy and the object hierarchy from the application and to provide a message with a type chain and an object chain, both in the parent-child direction. *See id.*, *e.g.*, page 8, lines 13-16; page 12, lines 24-27. The type chain includes a type node associated with a target object and the object chain includes an object node associated with the target object, and a combination of the type node and the object node identifies the target object that corresponds to the target component. *See id.*, *e.g.*, page 12, lines 24-27. The combination of ascendants of the type node and ascendants of the object node correspond to parent components. *See id.*, *e.g.*, page 12, lines 28-30. Further, the second computer has a message interpreter (*such as* 102, page 8, lines 16-17) operable to parse both chains to provide identification of the target component with type and object, as well as identification of the parent components with types and objects. *See id.*, *e.g.*, page 8, lines 16-18; page 13, lines 16-18. Identification includes translating the parsed information from the first natural language to a different natural language using both the chains. *See id.*, *e.g.*, page 12, lines 1-7.

Independent claim 4 is directed to a method for identifying a target component (*such as* 1222, page 12, lines 17-18) in an apparatus that has components related in hierarchy. The method includes representing the components by corresponding objects, thereby relating the objects in both a hierarchy identifying types of components and a different hierarchy identifying information associated with objects, where the type hierarchy and the object hierarchy identify

the components in a first natural language. *See id.*, *e.g.*, page 8, lines 11-20. The method further derives a message from information for both the type hierarchy and the object hierarchy. *See id.*, *e.g.*, page 2, lines 25-27. The message includes a type chain and an object chain, both in a parent-child direction, where the type chain includes a type node associated with a target object and the object chain includes an object node associated with the target object, further wherein a combination of the type node and the object node identifies the target object that corresponds to the target component. *See id.*, *e.g.*, page 8, lines 13-16; page 12, lines 24-27. Also, the combination of the ascendants of the type node and the ascendants of the object node correspond to parent components of the target component. *See id.*, *e.g.*, page 12, lines 28-30. The method further includes parsing the message to provide identification of the target component with type and object as well as identification of the parent components with types and objects, where the identification includes translating information from the first natural language to a different natural language using both type and object information. *See id.*, *e.g.*, page 8, lines 16-18; page 12, lines 1-7; page 13, lines 16-18.

Independent claim 6 is directed to a combination of complementary first and second computer program products for use in a computer system to identify a target component (*such as* 1222, page 12, lines 17-18) of an apparatus that has a plurality of components related in a hierarchy, where both computer products have machine-readable medium storing instructions for causing data processing. *See id.*, *e.g.*, page 3, lines 3-9. The first computer program product includes instructions operable to represent the components by corresponding objects, and thereby to relate the objects in both a hierarchy identifying types of components and a different hierarchy identifying information associated with objects, wherein the type hierarchy and the object hierarchy identify the components in a first natural language. *See id.*, *e.g.*, page 8, lines 11-20. The first computer program product further includes instructions operable to derive a message from information for both the type hierarchy and the object hierarchy, the message including a type chain and an object chain, both in the parent-child direction, where the type chain includes a type node associated with a target object and the object chain includes an object node associated with the target object, and further wherein a combination of the type node and the object node identifies the target object that corresponds to the target component. *See id.*, *e.g.*, page 2, lines 25-27; page 8, lines 13-16; page 12, lines 24-27. The combination of the ascendants of the type

node and the ascendants of the object node correspond to parent components of the target component. *See id.*, *e.g.*, page 12, lines 28-30. The second computer program product includes instructions operable to parse the message to provide identification of the target component with type and object as well as identification of the parent components with types and objects, where the identification includes translating information from the first natural language to a different natural language using both type and object information. *See id.*, *e.g.*, page 8, lines 16-18; page 12, lines 1-7; page 13, lines 16-18.

Independent claim 7 is directed to a method for identifying a target component (*such as* 1222, page 12, lines 17-18) by a first computer run-time environment to a second run-time environment. *See id.*, *e.g.*, page 3, lines 17-19. The method includes representing a plurality of objects by the first run-time environment, relating the objects in both a hierarchy identifying types of components and a different hierarchy identifying information associated with objects, and wherein the type hierarchy and the object hierarchy identify the components in a first natural language. *See id.*, *e.g.*, page 8, lines 11-20. The method further includes deriving a message from information for both the type hierarchy and the object hierarchy, the message including a type chain and an object chain, both in the parent-child direction, wherein the type chain includes a type node associated with a target object and the object chain includes an object node associated with the target object, and further wherein a combination of the type node and the object node identify the target object that corresponds to the target component. *See id.*, *e.g.*, page 2, lines 25-27; page 8, lines 13-16; page 12, lines 24-27. The combination of the ascendants of the type node and the ascendants of the object node also correspond to parent components. *See id.*, *e.g.*, page 12, lines 28-30. The message is then forwarded to the second run-time environment. *See id.*, *e.g.*, page 13, line 15. The second run-time environment then parses the message to provide identification of the target component with type and object and identification of the parent components with types and objects, wherein the identification includes translating information from the first natural language to a different natural language using both type and object information. *See id.*, *e.g.*, page 8, lines 16-18; page 12, lines 1-7; page 13, lines 16-18.

(6) Grounds of Rejection to be Reviewed on Appeal

I. Whether claims 1-4 and 6-8 are unpatentable over U.S. Patent No. 6,895,586 to Brasher et al. (hereinafter "*Brasher*") in view of U.S. Patent No. 6,345,245 to Sugiyama et al. (hereinafter "*Sugiyama*").

II. Whether claim 5 is unpatentable over *Brasher* in view of *Sugiyama* in further view of Tanenbaum, Andrew S., "Computer Networks." Third Edition, Prentice Hall PTR, 1996; pages 630-643 (hereinafter "*Tanenbaum*").

(7) Argument

I. The Combination of *Brasher and Sugiyama* Fails to Teach or Suggest the Claimed Invention.

A. ***Brasher* Fails to Teach or Suggest a First Application Relating the Objects in Both an Object Hierarchy and a Type Hierarchy**

The combination of *Brasher* and *Sugiyama* fails to teach each and every element of the claimed invention. For example, claim 1 recites, in part, "a first computer operable to execute a first application in which objects represent corresponding components, wherein the first application relates the objects in both a hierarchy identifying types of components and a different hierarchy identifying information associated with objects." In particular, the first application relates objects in both a type hierarchy and an object hierarchy. In contrast, *Brasher* merely teaches a single hierarchical namespace that is implemented over multiple computers such as in an enterprise network. *See Brasher*, col. 3, line 61 – col. 4, line 7. Using this single hierarchical namespace, common attributes and attribute types for objects are defined for all computers in the system. *See id.* at col. 15, lines 61-66.

Thus, *Brasher* does not teach or suggest both an object hierarchy and a different type hierarchy. In fact, the portions of *Brasher* cited as teaching the element in previous Office Actions explicitly support the proposition that *Brasher* merely describes a single object hierarchy (the "namespace"). For instance, *Brasher* teaches that "[t]he namespace comprises a logical arrangement of objects, stored hierarchically," and that the "namespace and its objects…may be

shared with the one or more computer systems of the enterprise." *Id.* at col. 3, line 62 – col. 4, line 7 (emphasis added). Later, *Brasher* describes an agent namespace as "one branch of a hierarchical, enterprise-wide namespace" that "may comprise a plurality of agent namespaces as well as namespaces of other components." *Id.* at col. 11, lines 35-39. Specifically, "[e]ach individual namespace may store a plurality of objects" and "may comprise a branch of a larger, enterprise-wide namespace." *Id.* at col. 11, lines 39-41. The namespace can then "employ a simple hierarchical information model in which the objects are arranged hierarchically," where "each object in the hierarchy may include a name, a type, and zero or more attributes." *Id.* at col. 11, lines 44-48 (emphasis added). Stated differently, *Brasher* teaches that the namespace is an object hierarchy alone, with each object in the hierarchy capable of including certain attributes, including an object type. *Brasher* does not, however, teach a type hierarchy separate from the object hierarchy as recited in claim 1.

In another cited portion, *Brasher* teaches that a root object has child objects, which in turn may also have their own child objects. *See id.* at col. 13, lines 52-58. In addressing or identifying the objects, each object "may be referred to using paths…formed by concatenating the object names encountered when traversing the path from the root of the tree to a given object." *Id.* at col. 13, lines 59-62. Further, the attributes contained in each object may then be used to uniquely define a particular object in the tree. *See id.* at col. 13, line 65 – col. 14, line 11. As an example, FIG. 6 (included below) illustrates a root object 402 ("enterprise") with two children objects ("hosts" 404 and "employees" 406). *See id.* at FIG. 6; col. 13, lines 52-56. The hosts object 404 in turn has two children objects of its own ("saturn" 408 and "neptune" 410).

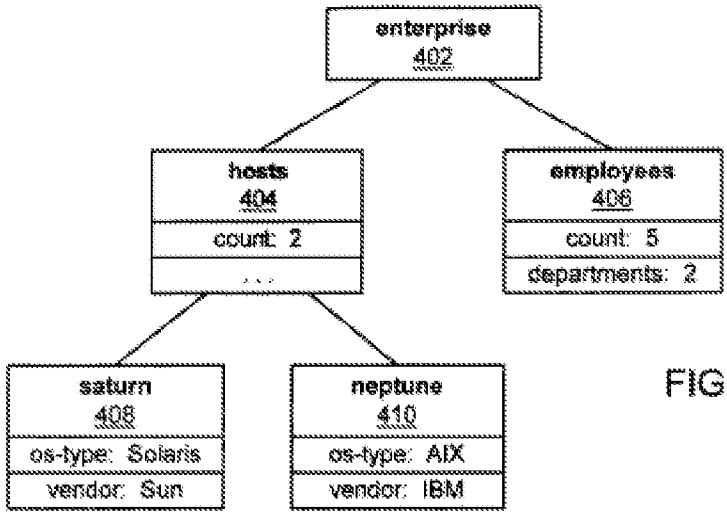

FIG. 6

As shown, each object contains at least one attribute which can be "referred to using a path and an attribute name." *Id.* at FIG. 6; col. 14, lines 5-6. For example, the "vendor" attribute of the "saturn" object 408 can be defined or referred to by the path "hosts/saturn vendor." *Id.* at col. 14, lines 6-7. In other words, *Brasher* teaches that each of its namespaces are object hierarchies that contain individual objects storing information that identify both object and type information for the particular object. Nowhere, however, does *Brasher* teach or suggest including both a type hierarchy and a different object hierarchy as recited in claim 1.

For at least these reasons, *Brasher* fails to teach or suggest each and every claim element of independent claim 1. Appellant respectfully submits that *Sugiyama* fails to account for at least the identified deficiencies of *Brasher*, and, accordingly, respectfully requests reconsideration and allowance of claim 1 and its dependents. Further, independent claims 4, 6, and 7 recite limitations that are similar, although not identical, to the element of claim 1 discussed above. Therefore, these claims and their dependents are allowable for at least reasons analogous to those discussed above in connection with claim 1.

B. **The *Brasher-Sugiyama* Combination Fails to Teach or Suggest a Translating Information From a First Natural Language to a Second Natural Language Using Both a Type and Object Chain**

Independent claim 1 recites that the "type hierarchy and the object hierarchy [of the first application] identify the components in a first natural language," and "wherein the second computer has a message interpreter operable to parse both chains to provide identification of the target component with type and object as well as identification of the parent components with types and objects, wherein the identification includes translating information from the first natural language to a different natural language using both the chains." For the teaching of this element, previous Office Actions merely identify the common dictionary management system disclosed in *Sugiyama*. *See, e.g.*, Office Action mailed February 5, 2008 at ¶ 5.

Appellant submits that the previous Office Actions have failed to consider each and every word of independent claim 1 in applying the rejection. As required by law, "[a]ll words in a claim must be considered in judging the patentability of that claim against the prior art." M.P.E.P. § 2143.03 (citing In re Wilson, 424 F.2d 1382, 165 U.S.P.Q. 494, 496 (C.C.P.A. 1970)). However, Appellant submits that the rejection fails to consider at least the claim 1 phrase reciting "wherein the identification includes translating information from the first natural language to a different natural language using both the chains." (emphasis added). More particularly, the previous Office Actions fail to consider that the translation from the first natural language to the second natural language uses both a hierarchy identifying types of components (a type hierarchy) and a hierarchy that identifies components based on information associated with objects (an object hierarchy).

In particular, *Sugiyama* merely teaches that a common dictionary 1 is updated by a common dictionary management portion 2 in the event that edits are made to local user dictionaries. *See Sugiyama* at col. 5, lines 28-35; Summary of the Invention. In connection with updating the common dictionary, the common dictionary management portion 2 may update other natural language processing systems such as word processing, machine translation, character recognition, abstract creation, voice recognition, and voice synthesizing. *See id.* at

claim 2. In doing so, dictionary edits executed by one natural language processing system "can be shared among specific natural language processing systems under predetermined conditions." *Id.* at col. 6, lines 1-5. In other words, other processing systems may be automatically updated with updated translation information. Nevertheless, *Sugiyama* contains no teaching or suggestion of translating between two natural languages using both a type hierarchy and an object hierarchy as recited by claim 1. Instead, *Sugiyama* merely teaches a one-to-one translation between words. *See id.* at col. 8, lines 17-42. As an example, *Sugiyama* teaches that the word "Internet" is translated into the Japanese term "Intahnetto" based on a previously-defined link between the English and Japanese words. *See id.* Thus, the *Sugiyama* method clearly fails to teach translating between natural languages using both the type and object chains received by the message interpreter of the second computer as recited in claim 1.

For at least these reasons, the *Brasher-Sugiyama* combination fails to teach or suggest each and every claim element of independent claim 1. Accordingly, Appellant respectfully requests reconsideration and allowance of claim 1 and its dependents. Further, independent claims 4, 6, and 7 recite limitations that are similar, although not identical, to the element of claim 1 discussed above. Therefore, these claims and their dependents are allowable for at least reasons analogous to those discussed above in connection with claim 1.

C. ***Brasher* Teaches Away From the Proposed Combination**

For the sake of argument, even if *Brasher* taught both a type hierarchy and a different object hierarchy, and even further if *Sugiyama* taught translating information from a first natural language to a second natural language using both the type hierarchy and the object hierarchy, nevertheless, there is no suggestion or motivation to combine *Brasher* and *Sugiyama*. In fact, *Brasher* teaches away from translating the objects within its namespace from one natural language to another, regardless of the method used to perform the translation.

First, *Brasher* does not mention different natural languages, much less translating between different natural languages. At best, *Brasher* states that its namespace is a logical organization that can be "physically located in the same room, in different countries, or anywhere in between." *Brasher*, col. 2, lines 14-17. To support the combination, the Advisory

Action speculates that "an international environment is interpreted as motivation to consider the possibility of a need to perform translations, motivating one of ordinary skill in the art to combine *Brasher* with...*Sugiyama*." Advisory Action mailed June 2, 2008, page 2. However, "[a] prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention." *W.L. Gore & Associates, Inc. v. Garlock, Inc.*, 721 F.2d 1540, 220 U.S.P.Q. 303 (Fed. Cir. 1983), cert. denied, 469 U.S. 851 (1984). (M.P.E.P. § 2141.02). Accordingly, Appellant submits that the Examiner has failed to consider *Brasher* in its entirety, and that *Brasher* teaches away from the proposed combination with *Sugiyama*.

Specifically, *Brasher* teaches that each object in a particular namespace is "uniquely addressable." *Brasher*, col. 11, lines 24-27. "'Uniquely addressable' refers to the property that items in a namespace have unique names such that any item in the namespace has a name different from the names of all other items in a namespace." *Id.* at col. 11, lines 27-30. Further, *Brasher* teaches that those unique names are used to identify a particular object in the namespace. *See id.* at col. 13, lines 59-60. For example, each object in the namespace is referred to using paths formed by concatenating the object names encountered when traversing the path from the root of the namespace down to a given object. *See id.* at col. 13, lines 59-62. If, as the Advisory Action suggests, by the *Brasher-Sugiyama* combination, the unique names of the particular individual objects within the namespace are translated between natural languages, the method of identifying objects in the namespace described by *Brasher* would no longer be available, and the "uniquely addressable" quality of each object providing each computer system with a unique name would be destroyed. In other words, the proposed combination of *Brasher* with *Sugiyama* would render the invention of *Brasher* unsatisfactory for its intended purpose. *See In re Gordon*, 733 F.2d 900; M.P.E.P. § 2143.01. For at least these reasons, the *Brasher-Sugiyama* combination is improper, and Appellant respectfully requests reconsideration of the 35 U.S.C. § 103(a) combination.

II. The Combination of *Brasher* and *Sugiyama* in further view of *Tanenbaum* Fails to Teach or Suggest the Claimed Invention.

The final Office Action rejects claim 5 under 35 U.S.C. § 103(a) as being unpatentable over *Brasher* in view of *Sugiyama* in further view of *Tanenbaum*. Appellant traverses these rejections and all findings and assertions therein. In particular, claim 5 depends from independent claim 4. As discussed above, independent claim 4 is allowable over the combination of *Brasher* and *Sugiyama*. The previous Office Actions have failed to cite any teaching or suggestion in *Tanenbaum* of the missing elements discussed above. Therefore, claim 5 is allowable at least because it depends from allowable claim 4. Thus, Appellant respectfully requests that the rejection of claim 5 be withdrawn.

CONCLUSION

In view of the foregoing, Appellant respectfully submits that the present claims are allowable over the cited references, and that the Examiner's rejection should be reversed. Should the Board be of the opinion that a claim on appeal may be amended to overcome a specific rejection, the Board is respectfully requested to include in the opinion such a statement and afford appellant the right to amend in conformity therewith.

The fee set forth in 37 CFR § 41.20(b)(2) in the amount of $510 is being paid concurrently herewith on the Electronic Filing System (EFS) by way of Deposit Account Authorization. Please apply any other required fees or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: August 13, 2008

/Michael E. Cox/
Michael E. Cox
Reg. No. 47,505

PTO Customer No. 32864
Fish & Richardson P.C.
1717 Main Street, Suite 5000
Dallas, Texas 75201
Telephone: (214) 760-6107
Facsimile: (877) 769-7945

Appendix of Claims

1. A computer system for identifying a target component in an apparatus that has components related in a hierarchy, the computer system comprising:

a first computer operable to execute a first application in which objects represent corresponding components, wherein the first application relates the objects in both a hierarchy identifying types of components and a different hierarchy identifying information associated with objects, wherein the type hierarchy and the object hierarchy identify the components in a first natural language;

a second computer coupled to the first computer via a network;

wherein the first computer includes a message generator operable to receive information relating to both the type hierarchy and the object hierarchy from the application and to provide a message with a type chain in a parent-child direction and an object chain also in the parent-child direction, wherein the type chain includes a type node associated with a target object and the object chain includes an object node associated with the target object, a combination of the type node and the object node identify the target object that corresponds to the target component, and a combination of ascendants of the type node and ascendants of the object node correspond to parent components; and

wherein the second computer has a message interpreter operable to parse both chains to provide identification of the target component with type and object as well as identification of the parent components with types and objects, wherein the identification includes translating information from the first natural language to a different natural language using both the chains.

2. The computer system of claim 1, wherein the first computer presents type-object hierarchy information to a first user and thereby adds type statements in a first language, and wherein the second computer presents identification of types in a second language.

3. The computer system of claim 1, wherein the message generator at the first computer is operable to append an identifier type to the type chain, and to append an identifier object to the object chain.

4. A method for identifying a target component in an apparatus that has components related in hierarchy, the method comprising:

representing the components by corresponding objects, thereby relating the objects in both a hierarchy identifying types of components and a different hierarchy identifying information associated with objects, wherein the type hierarchy and the object hierarchy identify the components in a first natural language;

deriving a message from information for both the type hierarchy and the object hierarchy, the message with a type chain in a parent-child direction and an object chain also in the parent-child direction, wherein the type chain includes a type node associated with a target object and the object chain includes an object node associated with the target object, a combination of the type node and the object node identify the target object that corresponds to the target component, and a combination of ascendants of the type node and ascendants of the object node correspond to parent components; and

parsing the message to provide identification of the target component with type and object as well as identification of the parent components with types and objects, wherein the identification includes translating information from the first natural language to a different natural language using both type and object information.

5. The method of claim 4, further comprising displaying the identification of the target component with type statements, wherein the type statements are provided locally.

6. A combination of complementary first and second computer program products, for use in a computer system to identify a target component of an apparatus that has a plurality of components related in hierarchy, both computer program products having a machine-readable medium storing instructions for causing data processing, instructions are distributed in the system as follows:

in the first computer program product to control a first computer, the instructions operable to represent the components by corresponding objects, and thereby to relate the objects in both a hierarchy identifying types of components and a different hierarchy identifying information associated with objects, wherein the type hierarchy and the object hierarchy identify the components in a first natural language;

in the first computer program product, the instructions further operable to derive a message from information for both the type hierarchy and the object hierarchy, the message with a type chain in a parent-child direction and an object chain also in the parent-child direction, wherein the type chain includes a type node associated with a target object and the object chain includes an object node associated with the target object, a combination of the type node and the object node identify the target object that corresponds to the target component, a combination of ascendants of the type node and ascendants of the object node correspond to parent components; and

in the second computer program product to control a second computer, the instructions operable to parse the message to provide identification of the target component with type and object as well as identification of the parent components with types and objects, wherein the identification includes translating information from the first natural language to a different natural language using both type and object information.

7. A method for identifying a target component by a first computer run-time environment to a second run-time environment, the method comprising:

representing a plurality of objects by the first run-time environment, thereby relating the objects in both a hierarchy identifying types of components and a different hierarchy identifying information associated with objects, wherein the type hierarchy and the object hierarchy identify the components in a first natural language;

deriving a message from information for both the type hierarchy and the object hierarchy, the message with a type chain in a parent-child direction and an object chain also in the parent-child direction, wherein the type chain includes a type node associated with a target object and the object chain includes an object node associated with the target object, a combination of the type node and the object node identify the target object that corresponds to the target component, a combination of ascendants of the type node and ascendants of the object node correspond to parent components;

forwarding the message to the second run-time environment; and

parsing the message by the second run-time environment to provide identification of the target component with type and object as well as identification of the parent components with types and objects, wherein the identification includes translating information from the first natural language to a different natural language using both type and object information.

8. The method of claim 7, wherein first and second run-time environments use different object models.

Evidence Appendix

NONE

Related Proceedings Appendix

NONE